# ALGORITHMS AND DATA STRUCTURES

| | | | | |
|---|---|---|---|---|
| **Faculty of Computer Science** | | | | |
| Study programme: | **Computer Science** | | Degree level: | **Engineer's degree full-time programme** |
| Specialization | **---** | | Diploma path: | **2026/2027W - 2026/2027S** |
| Module name: | **Algorithms and Data Structures**<br>( Algorytmy i struktury danych) | | | |
| Module type: | **obligatory** | **Semester: 1** | ECTS:**6** | Module ID:**FCS-00020** |
| No. of hrs in semester: | Lecture (L) - **30** Classes(C) - **30** Specialization workshop (SW) - **30** Project (P) - **0** Laboratory classes (LC) - **0** Seminar (S) - **0** | | | |
| Prerequisites | Calculus ( FCS-00002), Discrete Mathematics ( FCS-00054), Programming Basics ( FCS-00031), Object Oriented Programming ( FCS-00012), | | | |

| | | | |
|---|---|---|---|
| Aims and objectives: | The aim of the course is to develop the ability to evaluate the efficiency of an algorithm and to design efficient solutions even before the algorithm implementation phase. The student will gain knowledge of: methods for determining/estimating the computational cost of algorithmic solutions, methods for designing efficient algorithmic solutions, methods for designing efficient data structures, and computationally hard problems. The student will acquire skills in: designing computationally efficient algorithms and data structures, evaluating the efficiency of applied solutions, identifying computationally hard problems, and applying approximate solutions to such problems. The course also aims to develop the ability to communicate effectively about engineering and scientific problems with representatives of other fields.<br><br>References to the SFIA standard:<br>Programming/software development PROG – level 3<br>Software design SWDN – level 2<br>Scientific modelling SCMO – level 4 | | |
| Forms of teching activities:: | lecture, classes, specialization workshop, | Assessment: | Evaluation must be relevant to the intended learning outcomes:<br><br>Lecture - written exam; exercise - test; Laboratory - assessment of project |
| Module content: | Lecture<br><br>Basic concepts (algorithm correctness, computational complexity of algorithms, orders of growth of functions)<br>Recursion as a technique for encoding algorithms<br>Algorithm design techniques: divide and conquer, greedy methods, dynamic programming<br>Implementation of priority queues (using heaps)<br>Implementations of dictionary data structures (trees and hash tables)<br>Graph algorithms and data structures for graph representation<br>Computational complexity classes (P, NP, NPC, NP-hard), examples of computationally hard problems<br>Backtracking algorithms<br>Approximation and heuristic algorithms<br>Other topics (text processing algorithms, computational geometry, universal design)<br><br>Classes (Exercises)<br><br>Calculating the time complexity of iterative programs, determining the order of complexity, finding efficient solutions to basic computational problems<br>Exercises on recursive algorithms and divide and conquer: algorithm design and complexity analysis<br>Solving computational problems using greedy techniques and dynamic programming<br>Exercises on efficient data structures (heaps, trees, and hash tables)<br>Solving graph problems<br>Exercises on backtracking techniques<br>Developing skills in recognizing computationally hard problems and applying approximation/heuristic solutions<br><br>Specialized Laboratory<br><br>Basic computational problems, algorithm optimization (comparison, design, implementation) in terms of time complexity<br>Solving computational problems involving recursion / divide and conquer<br>Design and implementation of efficient algorithms (dynamic programming and greedy techniques) for solving combinatorial optimization problems<br>Implementation of tree-based dictionary data structures<br>Solving graph problems | | |
| Teaching methods: | subject exercises, programming, case method, lecture problem, informative lecture, | | |

**Learning outcomes**

| Symbol | Specify min. 4, max. 8 learning outcomes in the following order: knowledge – skills – competence. Each learning outcome must be verifiable | Reference to the programme learning outcomes of education |
|---|---|---|
| LO1 | knows the fundamental concepts and notations used in the analysis of algorithms and data structures and describing computing problems decision-making, etc., which can be solved by a computer. | INF1_W01<br>INF1_W05 |
| LO2 | is able to assess the time and memory complexity of algorithms and data structures. He can compare different solustions to the same problem. | H1_W01<br>INF1_W05 |
| LO3 | knows the standard solutions of designing and implementation of algorithms and data structures, their properties and application areas. | INF1_U04 |
| LO4 | can propose and design or choose algorithms and structures data to effectively solve a given engineering or scientific. Able to estimate the complexity of the problem and identify. | INF1_U04 |
| LO5 | is able to verify the correctness of algorithms using basic methods formal and simulation. | INF1_W06 |
| LO6 | knows how to implement algorithms and data structures using high level programming languages | INF1_U04 |

| | | |
|---|---|---|
| LO7 | Taking into account sustainability principles when selecting efficient algorithms. | H1_K03 |

| No. of learning outcome | Methods of assessing the learning outcome | Type of teaching activities (if more than one) during which the outcome is assessed |
|---|---|---|
| LO1 | test in the context of tutorials, written exam | L, C |
| LO2 | test in the context of tutorials, written exam | L, C |
| LO3 | test in the context of tutorials, written exam | L |
| LO4 | credit specialist laboratory tasks | Sw |
| LO5 | test, written exam, credit specialist laboratory tasks | L,C,Sw |
| LO6 | credit specialist laboratory tasks, observation of work during the workshop | Sw |
| LO7 | Assessment of partial problem-solving tasks. | SW |

| | | | |
|---|---|---|---|
| Student's workload (in hours) | **1** - Participation in lectures | None | 30 |
| | **2** - Participation in auditorium exercises and specialist. laboratories. | None | 60 |
| | **3** - Preparation for auditorium exercises / specialist workshop. | None | 10 |
| | **4** - Implementation of problem tasks in the specialist workshop. | None | 20 |
| | **5** - Preparation for passing the exercises. | None | 14 |
| | **6** - Participation in the consultations related to auditorium exercises and specialist workshop. | None | 4 |
| | **7** - Exam preparation. | None | 10 |
| | **8** - Exam attendance. | None | 2 |
| | | **TOTAL:** | **150** |
| Quantitative indicators | Student's workload - activities that require direct teacher participation: (1)+(2)+(6)+(8) | 96 | **ECTS** |
| | | | 3.8 |
| | Student's workload connected with practical classes (2)+(3)+(4)+(5) | 104 | 4.2 |

| | |
|---|---|
| Basic references: | 1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, S. Clifford, Introduction to Algorithms, Massachusetts Institute of Technology Press, 2022<br>2. R. Neapolitan, K. Namipour, Bases of algorithms with examples in C++, Helion, Warszawa, 2006<br>3. A. V. Aho, J. E. Hopcroft, J. D. Ullman, The Design and analysis of computer algorithms, Addison-Wesley Publishing Company, 1974 |
| Further reading | 1. Banachowski, K. Diks, W. Rytter, Algorithms and Data Structures, WNT, Warszawa, 2006<br>2. A. Drozdek, D. L. Simon Data structures in C, WNT, Warszawa, 2003<br>3. N. Wirth, Algorithms + Data Structures = Programs, WNT, Warszawa, 2004 |

| | | | |
|---|---|---|---|
| Unit: | Department of Theoretical Computer Science | Lecturer/ instructor | dr Joanna Karbowska-Chilińska,dr inż. Krzysztof Ostrowski |
| Date of issuing the programme: | 30th March 2026 | Author of the programme: | dr Joanna Karbowska-Chilińska |

L - lecture, C - classes, LC - laboratory classes, P-project, SW - specialization workshop, S - seminar